

ULTRIX

Guide to the uucp Utility

Order Number: AA-MF03B-TE
June 1990

Product Version: ULTRIX Version 4.0 or higher

**digital equipment corporation
maynard, massachusetts**

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause of DFARS 252.227-7013.

© Digital Equipment Corporation 1987, 1990
All rights reserved.

© AT&T 1979, 1984. All rights reserved.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

digital	DECstation	ULTRIX
CDA	DECUS	ULTRIX Mail Connection
DDIF	DECwindows	ULTRIX Worksystem Software
DDIS	DTIF	VAX
DEC	MASSBUS	VAXstation
DECnet	MicroVAX	VMS
	Q-bus	VMS/ULTRIX Connection

Hayes is a registered trademark of Hayes Microcomputer Products, Inc.

MICOM is a registered trademark of MICOM Systems, Inc.

UNIX is a registered trademark of AT&T in the USA and other countries.

Contents

About This Manual

Audience	v
Organization	v
Conventions	vi

1 Overview of the uucp Utility

1.1 Transferring Files with the uucp Utility	1-1
1.2 Required Hardware Configurations	1-2
1.3 Required Software	1-3

2 Setting Up the uucp Utility

2.1 Setting Up the uucp Utility with uucpsetup	2-1
2.1.1 Configuring the Modems	2-1
2.1.2 Defining Outgoing Connections	2-2
2.1.3 Defining Incoming Connections	2-2
2.1.4 Refreezing the sendmail Configuration File	2-3
2.2 Setting Up the uucp Utility Manually	2-4
2.2.1 Configuring Modem Lines	2-4
2.2.2 Setting Up the Password File	2-5
2.2.3 Setting Up the Remote System File	2-5
2.2.4 Setting Up the Dial-Codes File	2-8
2.2.5 Setting Up the Device File	2-8
2.2.6 Setting Up the Command File	2-9
2.2.7 Setting Up Spool Subdirectories	2-10
2.2.7.1 Installing Subdirectories on New Systems	2-10
2.2.7.2 Installing Subdirectories on Old Systems	2-11
2.2.7.3 Adding Individual System Subdirectories at a Later Time	2-11

3 Maintaining the uucp Utility

3.1	Polling Remote Sites	3-1
3.2	Compacting uucp Spool Directories	3-3
3.3	Monitoring the Network	3-3
3.3.1	Obtaining a Snapshot of the uucp System	3-3
3.3.2	Obtaining the Status of the uucp Utility	3-4
3.3.3	Obtaining a Log of File Transfers	3-5
3.4	Adding and Deleting Incoming Connections	3-6
3.5	Adding and Deleting Outgoing Connections	3-6
3.6	Adding, Deleting, and Moving Modems	3-6
3.7	Adding Direct-Connect Lines	3-7
3.8	Cleaning Up Excess Files and Directories	3-8
3.9	Maintaining System Security	3-8
3.9.1	File Access Security with USERFILE	3-9
3.9.2	Login Security with USERFILE	3-10
3.9.3	Remote Execution Security	3-11
3.10	uucp Commands	3-12

4 Configuring the MICOM PAD for uucp and tip

4.1	Description of the MICOM PAD	4-1
4.2	Connecting the PAD	4-1
4.3	Setting Up the uucp Utility for the PAD	4-2
4.3.1	Setting Up Incoming uucp Lines	4-3
4.3.2	Setting Up uucp to Dial Out Over the PAD	4-4
4.4	Setting Up tip for Use with the PAD	4-5
4.4.1	Setting Up Outgoing tip Connections	4-5
4.4.2	Setting Up Incoming login Service	4-6

5 Troubleshooting the uucp Utility

5.1	Transmission Messages in the LOGFILE	5-2
5.1.1	Problems During the Connection Stage	5-2
5.2	Obtaining a Log of Administrative Errors	5-5

About This Manual

This guide describes how to set up and maintain the ULTRIX uucp utility (UNIX to UNIX copy program).

Audience

The audience for this guide is anyone setting up and maintaining the uucp utility. The guide assumes that:

- You, or a DIGITAL service representative, have checked the hardware to ensure that it is working properly.
- Your software contains the uucpsetup command for automatic set up of the uucp utility. Manual setup tasks may be performed on any version of the ULTRIX operating system.

Organization

This guide consists of the following chapters:

- | | |
|-----------|--|
| Chapter 1 | Overview of the uucp Utility
This chapter introduces the uucp utility. |
| Chapter 2 | Setting Up the uucp Utility
This chapter describes the background information and tasks required for setting up the uucp utility automatically with the uucpsetup command. It also describes how to set up the uucp utility manually for those who may be interested. |
| Chapter 3 | Maintaining the uucp Utility
This chapter describes how to maintain the uucp utility. It describes how to modify the uucp files, add and remove modem lines, and so forth. It also addresses system security issues. |
| Chapter 4 | Configuring the MICOM PAD for uucp and tip
This chapter describes the MICOM PAD and how to set up the uucp utility for use with it. |
| Chapter 5 | Troubleshooting the uucp Utility
This chapter addresses how to handle communications problems and lists the error messages. It also suggests corrective action. |

Conventions

The following conventions are used in this guide:

`cat(1)` Cross-references to the *ULTRIX Reference Pages* include the appropriate section number in parentheses. For example, a reference to `cat(1)` indicates that you can find the material on the `cat` command in Section 1 of the Reference Pages.

filename In examples, syntax descriptions, and function definitions, italics are used to indicate variable values; and in text, to give references to other documents.

user input This bold typeface is used in interactive examples to indicate typed user input.

system output This typeface is used in interactive examples to indicate system output and also in code examples and other screen displays. In text, this typeface is used to indicate the exact name of a command, option, partition, pathname, directory, or file.

A number sign is the default superuser prompt.

>>>
CPU n >>> The console subsystem prompt is two right angle brackets on RISC systems, or three right angle brackets on VAX systems. On a system with more than one central processing unit (CPU), the prompt displays two numbers: the number of the CPU, and the number of the processor slot containing the board for that CPU.

RETURN This symbol is used in examples to indicate that you must press the named key on the keyboard.

The uucp utility allows you to transfer data from one system to another, and to execute commands on a remote system. Connections using the uucp utility can handle data communication over a wider geographic area than a local area network and usually transmit the data through telephone connections.

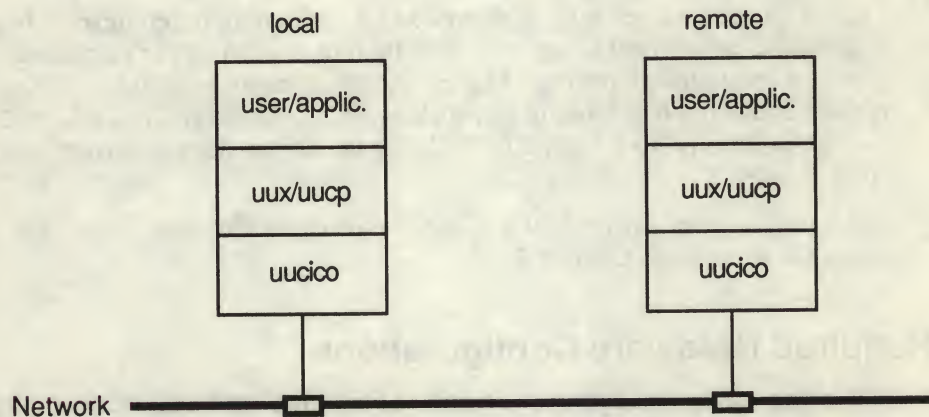
This chapter provides an overview of the uucp utility. It discusses the following topics:

- Transferring files with the uucp utility
- Required hardware
- Required software

For instructions on how to set up the uucp utility automatically using the uucpsetup command, see Chapter 2.

1.1 Transferring Files with the uucp Utility

The uucp utility system consists of three program levels:



ZK-0115U-R

At the highest logical level is the user or application program. This level uses the uux or uucp utility to initiate requests for both file transfers and remote job execution. The uucp utility spools user requests for file transfers. The uux program executes commands on a remote system. Each request is queued in the appropriate spool directory.

The uucico program performs the file transfer over the network. Before the transfer takes place, uucico must go through several stages.

First, the `uucico` program must call up the destination system and log in. After a successful login, the handshaking stage takes place between the destination `uucico` daemon process and the local daemon. At this stage, each daemon determines if the remote system has permission to use its local resources. If the handshake succeeds, then both daemons select the protocol to use to ensure that raw data is reliably transmitted over the network. Each daemon then uses the corresponding packet driver software to send and receive raw data.

The file transfer process then begins. The local daemon searches the spool directories for job requests, builds a list of files to transfer, and then starts transmitting the files. A file transfer protocol is used to ensure that each file is successfully transmitted only once. This protocol also notifies you if the `uucico` program cannot transmit the file and explains why the transfer failed. After the `uucico` program transfers all of the files, the destination site begins to transfer files back to the local system. When both systems have no more work to do, the connection through the network is broken, and the conversation is complete.

Throughout the conversation, temporary files are created and removed. For example, lock files are created in the top level spool directory `/usr/spool/uucp`. These lock files correspond to the remote system (`LCK..sys`) and the hardware device used for communication (for example, `LCK..cua0`).

The `uucp` utility includes the remote execution program `uux` and the remote execution daemon `uuxqt`. The `uux` and `uuxqt` programs execute commands on a specified remote system. The `uux` command spools the command and associated data into the appropriate spool directories. The `uucico` daemon then transfers the files to the remote system. At the remote system, `uuxqt` starts when these execution files arrive.

The `uuxqt` program scans through the execution spool directories `/usr/spool/uucp/sys/DEFAULT/X.` or `/usr/spool/uucp/sys/systemname/X.` searching for commands to execute. If a command has arrived along with the data files needed by the command, `uuxqt` tries to execute the command. The `uuxqt` program also creates LCK files for the type of command it is looking for, such as `LCK.XQTrmail` or `LCK.XQT`. The `uuxqt` program also creates a LCK file for the command it is currently working on, such as `LCK.X.chicagoX0123`.

Shell scripts are run periodically to remove outdated temporary files. For information about the scripts, see Chapter 3.

1.2 Required Hardware Configurations

The `uucp` utility can operate with any of the following hardware configurations:

- All Digital modems with Auto Call Units (ACU), as listed in the Software Product Description (SPD) included in your media kit
- A direct connection with a null modem cable such as a BC03-M
- A direct connection with a modem link

To connect the Digital modems and corresponding ACUs follow these steps:

1. Connect the modem to a port on the local system using a straight-through cable.
2. Connect the ACU to the phone line by following the instructions in the modem's User's Guide.

3. Set the ACU communications baud rate; see the switch options in the modem's User's Guide.

For the `tip` command to work, the modems must be set up properly for `uucp`. To install a hardwired direct link, connect a null modem cable from a port on the local system to a port on the remote system. When using a modem, connect the modem from a port on the local system to a port on the remote system with a straight-through cable. For more information, follow the modem's installation instructions.

1.3 Required Software

The `uucp` program requires these files and directories to run:

<code>/var/uucp/USERFILE</code>	File that defines <code>uucp</code> security
<code>/etc/passwd</code>	Password file
<code>/etc/acucap</code>	Automatic call unit capabilities file
<code>/var/uucp/L.sys</code>	Information needed to connect to a system
<code>/var/uucp/L-devices</code>	Devices used to connect to remote systems
<code>/var/uucp/L-dialcodes</code>	Dial code abbreviations
<code>/var/uucp/L.cmds</code>	Allowable remote execution commands
<code>/var/uucp/Makefile</code>	File that creates default spool directories
<code>/usr/spool/uucp/sys/*</code>	Directories for depositing spooled files and temporary work files
<code>/etc/ttys</code>	File used for setting up modem lines
<code>/etc/remote</code>	Remote host description file

Files specific to the `uucp` utility are set up automatically when you execute the `uucpsetup` command. For information about setting up the `uucp` utility, see Chapter 2.

1. The first part of the report is a general introduction to the subject.

2. The second part of the report is a detailed description of the methods used in the study.

3. Results and Discussion

The results of the study are presented in this section. The data show that there is a significant correlation between the variables studied. The discussion of the results is based on the findings of the study and compares them with the results of previous studies. The conclusion of the study is that the variables studied are related in a significant way.

The study was conducted in a laboratory setting and the results may not be directly applicable to the real world. Further research is needed to confirm the findings of this study.

This chapter describes how to set up the uucp utility on your system. The first section describes how to use the `uucpsetup` command, which is the recommended setup method. The second section describes how to set up the uucp utility manually.

2.1 Setting Up the uucp Utility with uucpsetup

This section describes how to set up the uucp facility using `uucpsetup`.

Note

If the Yellow Pages (YP) service is running on your system, read the *Guide to the Yellow Pages Service* for information on how to modify YP maps before you use the `uucpsetup` command.

To run the `uucpsetup` command, type:

```
# /etc/uucpsetup -a
```

The `-a` option tells `uucpsetup` that this is a first-time installation; the program will prompt you for information about setting up the modems and incoming and outgoing connections. In addition to setting up uucp, the `-a` option allows you to set up the modems for use with the `tip` command.

Once `uucpsetup` is running, it sets up uucp by doing the following:

1. Configures the modems
2. Defines the outgoing connections
3. Defines the incoming connections

The following sections describe these steps.

2.1.1 Configuring the Modems

The `uucpsetup` command presents this prompt:

```
How many modems are you adding to this system [1] ?
```

Supply the number of modems you are adding to the system. You are then asked to supply the modem type, such as `df224`, the tty line, such as `tty25`, and whether each tty modem line is incoming, outgoing, or shared. A shared tty line is for both incoming and outgoing connections.

After you have supplied the modem information, `uucpsetup` gives you the opportunity to supply the information again, exit, or continue. The modems must be set up for the `tip` utility to work, regardless of whether your site will be using uucp.

Note

Remember to connect the modems physically to your system either before you run `uucpsetup`, or directly afterwards.

2.1.2 Defining Outgoing Connections

During this step, you answer questions to define the calls your system will make to remote systems (outgoing connections). For each outgoing connection, you specify:

- The system name. This is the full name of the remote system, as defined when it was installed.
- The time to call. Specify when your system is allowed to call the remote system. The options are:
 - any time of any day
Sunday through Saturday, 24 hours a day.
 - evenings
Monday through Friday, 5:00 P.M. to 8:00 A.M.; all day Saturday and Sunday.
 - nights
Monday through Friday, 11:00 P.M. to 8:00 A.M.; all day Saturday; Sunday until 5:00 P.M.
 - never
Use this option for systems that can call your system, although your system never calls them.

Specifying this information only lists the times your system is allowed to call a remote system. It does not cause periodic polling.

- The line speed of the remote system. Enter the baud rate expected by the remote system. The default rate is 1200 bits per second.
- The telephone number of the remote system. An equal sign (=) in a phone number tells the modem to wait for a second dial tone before dialing the rest of the numbers.
- The login name. You specify the login name to be passed to the remote system for this system's login.
- The password. You specify the password to be passed to the remote system for this system's login.

After you have supplied the outgoing `uucp` information for each system, `uucpsetup` gives you the opportunity to retype the information.

This series of outgoing connection questions repeats until you press the RETURN key in response to the request for the next system name.

2.1.3 Defining Incoming Connections

In this step, `uucpsetup` prompts you to define the remote systems that can access your system.

Before `uucpsetup` prompts for information about the remote systems, it creates two standard entries: one called *local* and one called *remote*. These two entries define the default access for the local system and for all the remote systems that do

not have specific entries, if an INSECURE file exists. For further information, see Section 3.9.1.

The local entry defines the privileges extended to all users with an account on the local system. These users are given the highest execution access level (9) and can access any directory allowed by the file protection scheme in the ULTRIX system. For more information about execution access levels, see Section 3.9.2.

The remote entry defines the privileges extended to all remote systems not otherwise defined in the uucpsetup dialog. These remote systems are allowed to execute a limited number of commands on your system, such as rmail. Also, these systems can transfer files to and from your system, but only to and from the /usr/uucp/uucppublic directory. The execution access level of remote systems using the default remote entry is defined as 1.

For each incoming connection, you specify:

- The system name. This is the full name of the remote system, as defined when it was installed.
- A short comment for the password file. The text you enter here is entered into the password file entry for the system. It is a good idea to enter the site's location, company name, or some other form of identification.
- The password for the remote system. This is the password expected from the remote system when it attempts to log in to the local system.
- The execution access level for the remote system. A number from 0 through 9, defining the commands that can be executed by the remote system. For further information, see Section 3.9.2.
- The callback option. To optimize your system's security, you can elect to call back this remote system when it attempts to call. In that way, you can ensure that you are making the connection with the remote system only. Otherwise, someone who knew the remote system's name and password could log in to your system without authorization.

Note

Be aware that if you select the callback option, you pay the telephone bill for the remote connection.

- The directory path for the remote system. This defines the directory from which the remote system can copy files. The default is the directory /usr/spool/uucppublic.

This series of incoming connection questions repeats until you press the RETURN key in response to the request for the next system name.

2.1.4 Refreezing the sendmail Configuration File

Because the sendmail program reads the /var/uucp/L.sys file only at freeze time, you must refreeze the configuration file after you add uucp sites. If you do not refreeze the file, you will not be able to exchange mail with new sites. After you update the files using the uucpsetup -i command, bring the system to single-user mode, and specify your system's host name, you can refreeze the sendmail configuration file.

Use the following command to refreeze the sendmail configuration file:

```
# /usr/lib/sendmail -bz
```

2.2 Setting Up the uucp Utility Manually

This section discusses how to set up the uucp utility manually. Note that the recommended method of setting up the uucp utility is with the uucpsetup command, described in Section 2.1. This section discusses:

- Configuring modem lines
- Setting up the password file
- Setting up the remote system file
- Setting up the L-dialcodes file
- Setting up the device file
- Setting up the command file
- Setting up spool subdirectories

2.2.1 Configuring Modem Lines

Before configuring the modem lines, you need to know the following information:

- The type of modem you are using
- The baud rate of your modem
- The device special file associated with the modem line that will be connected
- Whether your modem handles dial-out service only, dial-in service only, or both, through shared lines

To configure the modem lines for use with the uucp utility, follow these steps:

1. Edit the `/var/uucp/L-devices` file. Add an entry for the modem device special file and modem type. Here is an example of an L-devices entry:

```
ACU   ttyd5   ttyd5   2400   df224
```

2. Change the name and mode of the special device. By convention, modem lines are named `ttydn`, with *n* representing a unique modem number. For this reason, note that the modem line `tty05` is moved to `ttyd5` in the following example:

```
# cd /dev
# mv tty05 ttyd5
# chmod 666 ttyd5
```

If the modem line will be used for incoming calls only, change the mode to 622 instead of 666.

3. Edit the `/etc/ttys` file. Modify the entry for the appropriate line to turn on modem control. For example:

```
ttyd5 "/etc/getty std.2400" dialup on modem shared #oldname=tty05, df224
```

This entry allows the modem to be called and to call out. If you do not want to allow incoming calls, specify `off` instead of `on`.

4. Activate the changes to the `ttys` file as follows:

```
# kill -HUP 1
```

If you are setting up the modem for use with the `tip` command, you also need to edit the `/etc/remote` file. Modify the appropriate entry for the modem device name and type. Here is an example of an entry for a 2400-baud modem:

```
dial2400:dv=/dev/ttyd5:br#2400:at=df224:du:
```

For further information about the `/etc/remote` file, see `remote(5)` in the *ULTRIX Reference Pages*.

2.2.2 Setting Up the Password File

Any system that connects to the local system must have an entry in the `/etc/passwd` file. The system manager at each installation must agree on a login name and password, and this information must be entered into the password file. For information about the password file, see the *Guide to System Environment Setup*. Note that the last two fields of the entries for the connecting systems are as follows:

```
/usr/spool/uucppublic:/var/uucp/uucico
```

The last field is the path of the program which transfers the files, `uucico`. When a remote system logs in successfully, `uucico` is run in slave mode, and the dialog between the peer `uucico` programs begins.

Note

After installing the ULTRIX operating system, make sure you do not change the user identification number (UID) of the `uucp` password entry by overwriting your `/etc/passwd` file with a previous version of the file.

Before you restore your previous `/etc/passwd` file, compare it against the current version. If there are differences, edit the old `/etc/passwd` file and delete the lines that contain conflicting entries. You can then merge the two files. For example, if your previous file is called `/etc/passwd.old`, to merge the two files, type:

```
# cat passwd.old >> /etc/passwd
```

If Yellow Pages is running on your system, refer to the *Guide to the Yellow Pages Service* for more information on setting up the `/etc/passwd` file.

2.2.3 Setting Up the Remote System File

The `L.sys` file contains entries for each remote system that the local system can call. More than one line can be present for a particular system. In this case, the additional lines represent alternative communication paths that are tried in sequential order.

The format of each entry, with each field separated by blanks or tabs, is:

system-name time device class phone login_sequence

system-name The name of the remote system.

time A string that indicates the days of the week and times of day when the system can be called (for example, MoTuTh0800-1740).

The day portion can be a list containing one or more of the following:

Su Mo Tu We Th Fr Sa

The day portion can also be Wk for any weekday or Any for any day.

You can indicate hours in a range (for example, 0800-1230). If you do not specify a time portion, calls are allowed at any time.

You can specify a time range that spans 0000.

For example, 0800-0600 means that times from 8 a.m. to the following 6 a.m. are allowed. It also means that times from 6 a.m. to 8 a.m. are not allowed. You can specify multiple date specifications with vertical bars (|).

For example, Wk0100-0600 | Sa | Su means that the system can be called any weekday between 1 a.m. and 6 a.m. or any time on Saturday or Sunday.

An optional field is available to indicate the minimum time (in minutes) that must elapse before retrying a failed connection. A failed connection attempt is a login failure, as opposed to a dialing (connection) failure. The field separator is a comma (,). For example, Any, 9 means that your site can call any time, but that it must wait at least 9 minutes after a failure has occurred.

device

Either the ACU or the hardwired device used for the call. For the hardwired device, use the last part of the special file name, such as tty02.

class

The line speed for the call, for example 1200.

phone

The phone number, made up of an optional alphabetic abbreviation and a numeric part. The abbreviation should be one that appears in the L-dialcodes file, for example, ct5900, nh6511, or an unabbreviated phone number. For the hard-wired devices, this field contains the same string as used for the *device* field.

login_sequence

The login information, given as a series of fields in this format:

expect[-sendspecial-expect] send ...

The *expect* field is the string expected to be read when logging in to the remote system, and *send* is the string to be sent when the *expect* string is received. If *expect* is not received, the field can be set up so that special characters (*sendspecial*) can be transmitted to the remote site. After the special characters are transmitted, the *expect* following *sendspecial* is the next expected string.

Two special characters can be sent when *expect* is not received: EOT and BREAK. EOT sends an EOT character, and BREAK sends three break sequences, simulated with line speed changes and null characters. A number from 1 to 9 can follow BREAK. For example, BREAK1 sends one break. Note that after every send string a carriage return is sent, except as noted below. If *sendspecial* is two consecutive dashes (--), a carriage return is sent.

Other special characters can be used as part of the send sequence for remote systems that are slow or that need to be placed in a state

before the true login sequence can begin. These are the special characters and their meanings:

PAUSE <i>n</i>	Pause <i>n</i> seconds, or five seconds by default.
\d	Pause one second before sending the next character.
\s	Send a blank character.
\r	Send a carriage return.
\c	Do not send a carriage return at the end of the send string.
\b	Send a break character.
\###	Send the ASCII character represented by octal number ###. For example, \05 is CTRL/E.
P_ZERO	Change the parity from even (default) to zero.
P_EVEN	Change the parity to even.
P_ODD	Change the parity to odd.
P_ONE	Change the parity to one parity.

In some instances, it is necessary to send characters to the remote system before expecting something to arrive. For example, some systems expect a carriage return before issuing a login prompt. A sequence of two double quotation marks ("") used as the *expect* string indicates that nothing is expected, and the following send string is transmitted:

```
"" \r\c
```

This sequence expects nothing, and then sends a carriage return. Using the \c means the default \r should not be sent. If the \c were not here, two carriage returns would be transmitted.

The following examples show alternative expect-send sequences.

```
ogin: xuucp ssword: foobaz
```

The string *ogin:* is expected. When it is received, *xuucp* is sent. Now the word *ssword* is expected. The first letter of "password" varies from system to system, so it is safer to look for the tail end, for example, *ssword*. When *ssword* is received, *foobaz* is sent. If the login is successful, the conversation between the peer transfer processes (*uucico*) begins. If the login fails, the connection attempt fails.

```
ogin:--ogin: xuucp ssword: foobaz
```

The string *ogin:* is expected. If it is not received, then send a carriage return and expect *ogin:* again. If *ogin:* is received, send *xuucp*. Then, the procedure in the previous example is followed.

```
ogin:-BREAK1-ogin: xuucp ssword: foobaz
```

The string *ogin:* is expected. If it is not received, send one break sequence to change the baud rate of the remote *getty* process, and expect *ogin:* again. Then proceed as in the previous examples.

The following examples illustrate the use of these special characters.

```
"" @ login: xuucp ssword: foobaz
```

Expect no strings, send an at sign (@) character (line kill), followed by a carriage return, and then continue the normal login sequence. The at sign character is often useful for clearing out line noise before starting to log in. The default line kill character varies from system to system.


```
" " P_ZERO " " \d\b " " \005\c login: xuucp
ssword: foobaz
```

Expect no strings, change output parity to zero parity and expect nothing. Wait one second, and then send a break sequence, expect nothing, and send the escape character \005 without the default carriage return. Then continue with the normal login sequence.

Putting all the fields together, the following examples illustrate complete L.sys entries. Assume each entry is on one line:

```
sys1 Any ACU 1200 wy7777 " " \r\d ogin:-EOT-ogin: Ufoobaz
ssword: secret
```

```
sys1 Any ttyae 9600 tty32 " " \r\d ogin:-EOT-ogin: Ufoobaz
ssword: secret
```

```
sys2 Any ACU 300 456-1234 " " \r ogin:-EOT-ogin:-EOT-ogin:
Ussystem1 ssword: testing
```

```
sys3 Any,10 ACU 1200 8=123456789 ogin:-\r\c-ogin: \
-\r\c-ogin:-BREAK-ogin:-EOT-ogin: Ussystem1 ssword: huskies
```

```
sys4 Any0000-0700 ACU 1200 8=987654321 ogin:-BREAK-ogin: \
-BREAK-ogin:-BREAK-ogin: Ussystem2 ssword: froglegs
```

If the remote system uses nonstandard hardware, the L.sys entry can become complex. To connect to some systems, you may have to alter the L.sys entries until a successful combination is found.

Note

The L.sys file must also contain an entry for the name of any system that calls you, but that you do not call. The form of such an entry is abbreviated to the system name and one word in place of the time entry, such as never or incoming.

2.2.4 Setting Up the Dial-Codes File

The L-dialcodes file contains the dial-code abbreviations used in the L.sys file, for example, nh, which could expand to the area code for New Hampshire, USA.

The entry format is:

abb dial-seq

abb

The abbreviation as used in the L.sys file.

dial-seq

The dial sequence to call that location.

For example, the entry nh 603 would force any L.sys entry that used the prefix nh in the *phone* field to send 603 to the dial unit before the rest of the phone number is dialed.

2.2.5 Setting Up the Device File

The device file L-devices contains information about call units and direct connections. It is used to map specifiers in the L.sys file to specific devices.

The format for each entry is:

type line call-unit speed brand preferred-proto

type

A device type, such as ACU or DIR. DIR indicates that this is a direct connect, hardwired line.

line

The device for the modem line or hardwired line as named in /dev, such as ttyd1.

call-unit

The automatic call unit associated with *line*, such as cua0. Hardwired lines should place the device for the line in this field, for example, ttyd1. Because call units are rarely used, the value for *call-unit* is usually the same as the value for *line*.

speed

The baud rate.

brand

The brand name of the modem. The acceptable brands are any of the Digital modems discussed in Section 1.2, as well those modems you have configured for your system in the /etc/acucap file. See acucap(5) in the *ULTRIX Reference Pages* for more information. For direct connections, place the word *direct* in this field.

preferred-proto

The protocol you prefer. The *g* protocol is the standard protocol for the uucp program. The *f* protocol is designed to work over the MICOM PAD. For information about the MICOM PAD, see Chapter 4.

Here are some typical L-devices entries:

ACU	ttyd1	ttyd1	300	df02	g
ACU	ttyd2	ttyd2	1200	df03	g
ACU	ttyd3	ttyd3	1200	df112	g
DIR	tty04	tty04	9600	direct	f

The *g* in these entries is the specified default. If the protocol is not specified, the system defaults to the *g* protocol.

2.2.6 Setting Up the Command File

The command file *L.cmds* contains the list of commands that a remote system can execute by means of the *uux* command. The format of the *L.cmds* file is:

command Xn

command

A command or application program.

Xn

The execution level associated with *command*. The number can range from 0 through 9. If the *X* field is not present, then 9 is provided as the default level. If *X* is present but a level number is not specified, then 0 is assumed, enabling any system to execute this command. You should limit the number and type of allowable commands appropriately.

A typical list of commands might be:

```
rmail X1
rnews X1
uusend X1
```

To specify the path that `uuxqt` uses to search for commands, add the following line anywhere in the `L.cmd` file:

```
PATH=path1:path2:path3:path4:...
```

For example:

```
PATH=/bin:/usr/bin:/usr/ucb
```

In this example, the `uuxqt` daemon first searches `/bin` for a command; if the command is not in `/bin`, it then searches `/usr/bin` and then `/usr/ucb`. If no `PATH` entry is supplied, the default `PATH=/bin:/usr/bin` is used.

2.2.7 Setting Up Spool Subdirectories

These subdirectories must be created for various `uucp` files:

<code>/usr/spool/uucp/TM.</code>	Temporary files
<code>/usr/spool/uucp/STST.</code>	Connection status files
<code>/usr/spool/uucp/sys</code>	System subdirectories

The `sys` subdirectory is further divided into directory names that correspond to specific remote systems and a default directory (`DEFAULT`). The minimum configuration requires a `/usr/spool/uucp/sys/DEFAULT` directory. However, you must create individual system subdirectories manually. The following command creates the subdirectories for a system named `system1`:

```
# /var/uucp/uumkspool system1
```

Each system directory contains these subdirectories:

<code>D.localname</code>	Outgoing data files
<code>D.localnameX</code>	Outgoing remote execution files
<code>D.</code>	Incoming data files
<code>C.</code>	Work files
<code>X.</code>	Incoming remote execution request files

If a large amount of traffic is expected between specific systems, then you should create subdirectories for those systems. If you have to create new system directories, you must move files out of the `DEFAULT` directory and into the new system spool directories. For more information, see Section 2.2.7.3. The following sections explain how to install subdirectories.

2.2.7.1 Installing Subdirectories on New Systems – The `/var/uucp/Makefile` file includes shell scripts to create the needed subdirectories. To create the minimum set of directories, enter the `/var/uucp` directory and type:

```
# make mkdirs
```

Note that some directories may already exist.

If this command does not succeed, use the following commands, where *system1* is the uucp name of the local system:

```
# cd /usr/spool/uucp
# csh
# mkdir TM. STST. sys
# chown uucp TM. STST. sys
# chmod 0755 TM. STST. sys
# cd /usr/spool/uucp/sys
# mkdir DEFAULT
# chown uucp DEFAULT
# chmod 0755 DEFAULT
# cd DEFAULT
# foreach i (C. D.system1 D.system1X D. X.)
? mkdir $i
? chown uucp $i
? chmod 0755 $i
? end
#
```

To create individual system subdirectories, use this format:

```
/var/uucp/uumkspool sys1 sys2 ...
```

The sys fields are the names of the system subdirectories.

All of the required subdirectories should now exist.

2.2.7.2 Installing Subdirectories on Old Systems – If subdirectories are to be installed on a system that has been running with a different subdirectory scheme, follow the steps in Section 2.2.7.1.

Then, move the files from the old directory to the new subdirectories.

Use the `/var/uucp/uurespool` command to move old spool files to new spool directories. The `-tn` option lets you specify the type of spool that was used prior to installing the new uucp utility.

The number in the `-tn` option is 1, 2, 3, or 4:

- Use 1 if all spool files are located in `/usr/spool/uucp`. This is the format of the old uucp utilities. There are no subdirectories.
- Use 2 if the spool directory has been split out into several subdirectories: `D.local`, `D.localX`, `D.`, `X.`, and `C.`
- Use 3 if the spool directory has been split as in option 2 with the addition of `C./OTHERS` and `STST`.
- Use 4 if a new system directory has been created and the spool files have to be moved from `DEFAULT` to the new system directory.

For example, if the spool directory has been split out into `D.local`, `D.localX`, `X.`, and `C.`, type this command line:

```
# /var/uucp/uurespool -t2
```


2.2.7.3 Adding Individual System Subdirectories at a Later Time – If additional system subdirectories are to be added, the new directory needs to be created and spooled files moved from DEFAULT to the new directory. Type these commands:

```
# /var/uucp/uumkspool sys1 sys2 sys3  
# /var/uucp/uurespool -t4
```

In this example, sys1, sys2, and sys3 are system names.

This chapter lists some of the basic system management tasks for maintaining the uucp facility.

Note

If the Yellow Pages (YP) service is running on your system, read *Guide to the Yellow Pages Service* for information on how to modify YP maps before you use the uucpsetup command.

Maintaining and modifying the uucp utility is a complex process. Following are the basic tasks discussed in this chapter:

- Polling remote sites
- Compacting uucp spool directories
- Monitoring the network
- Adding and deleting incoming connections
- Adding and deleting outgoing connections
- Adding, deleting, and moving modems
- Adding multiple telephone numbers for outgoing connections
- Cleaning up excess files and directories
- Maintaining uucp security

This chapter describes briefly how to perform these tasks.

3.1 Polling Remote Sites

The uucpsetup command allows you to define the time of day that your system is allowed to make outgoing connections to remote sites. If a user requests a uucp transaction during the times your system is allowed to make outgoing connections, the connection is made immediately. However, if the user makes the request at a time when your system is not allowed to make outgoing connections, the request is queued. It remains queued until either the remote site calls your system, or another outgoing request is made from your system during a time when outgoing connections are allowed.

You might want to have your system periodically poll those remote sites that have calling time restrictions. If you set up polling, your system automatically calls the remote site at regular intervals. If the remote site has any queued requests for your system, then when your system polls the remote site, the queued requests are processed.

There are five intervals you can specify for polling remote sites:

hour Poll the remote site once every hour on the half hour.
 day Poll the remote site once a day at 6:00 A.M.
 noon Poll the remote site once a day at 12:00 noon.
 night Poll the remote site once a day at 2:00 A.M.
 longhall Poll the remote site when the phone rates are least expensive.

There are six shell scripts in the /var/uucp directory, one for each polling interval:

uucp.day	This script is run at the start of the day. It deletes any temporary files and saves the previous day's LOGFILE and SYSLOG data in LOGFILE.yesterday and SYSLOG.yesterday. It also contacts any systems listed in /var/uucp/LIST.DAY, and then starts a general poll of all systems for which there is work. Messages indicating the progress of this shell script are placed in /var/uucp/LOG.shells. You should modify LIST.DAY whenever necessary so that the correct systems are called. If LIST.DAY is empty or nonexistent, only the general poll is performed.
uucp.hour	This script initiates a general poll every hour. Messages indicating the start and finish of the poll are sent to the console. If any systems are listed in LIST.HOUR, those systems are contacted on an hourly basis.
uucp.noon	This script contacts any system listed in LIST.NOON at noon.
uucp.night	This script cleans up the spool directories in the early morning hours, using the uuclean command. By default, any spool files older than 168 hours are removed. You should adjust this time limit to conform to local conditions. After the cleanup, any systems listed in LIST.NIGHT are contacted. A general poll contacts any remaining systems for which requests are still queued. Shell script messages are sent to LOG.shells.
uucp.week	LOG.shells is saved in LOG.shells.week and a general poll is started.
uucp.longhall	This script calls remote systems that are listed in UUCP.LONGHALL. You can use this script to reduce your telephone bill by calling those systems that are very far away when the phone rates are the least expensive.

To have a remote site polled, add its name to the appropriate file. For example, to have a remote site named markie polled every day at noon, edit /var/uucp/LIST.NOON and add the name markie on a new line.

You should modify the /usr/lib/crontab file to make it execute these shell scripts at the appropriate intervals. The following are typical /usr/lib/crontab entries:

```
30 * * * * su uucpa < /var/uucp/uucp.hour
0 12 * * * su uucpa < /var/uucp/uucp.noon
0 6 * * * su uucpa < /var/uucp/uucp.day
0 1 * * 5 su uucpa < /var/uucp/uucp.week
45 2 * * * su uucpa < /var/uucp/uucp.longhall
```


Note that for your system to poll a site at a particular interval, it must be allowed to make outgoing connections at that time.

3.2 Compacting uucp Spool Directories

Periodically, it may be necessary to compact the spool directories. Type this command to compact the uucp spool directories:

```
# /var/uucp/uucompact
```

If the uucompact command is not available, use the following command sequence to pack a directory called *directory*:

```
# mkdir tempdir
# chown uucp tempdir
# mv directory/* tempdir
# rm -rf directory
# mv tempdir directory
```

In this example, *tempdir* is a temporary directory.

For more information on uucompact, see uucompact(8c) in the *ULTRIX Reference Pages*.

3.3 Monitoring the Network

Cleanup shell scripts executed from the crontab file keep a well-tuned network clean of any miscellaneous files and any files that cannot be transferred. However, there are times when the network starts to become backlogged. Two programs help you monitor the network for such problems: uumonitor and uustat.

3.3.1 Obtaining a Snapshot of the uucp System

The /var/uucp/uumonitor command creates a snapshot of the uucp system. Following is the format of the output:

```
system-name #C #X most_recent_status CNT:# time
```

<i>system-name</i>	The remote system for which the entry applies.
#C	The number of C.files queued for the remote system.
#X	The number of requests for remote execution from the remote system.
<i>most_recent_status</i>	The result of the most recent attempt to connect to the remote system.
CNT:#	The number of times that a failure to log in to the remote system has occurred. This does not include the number of failed dial attempts.
<i>time</i>	The time the last status entry was made for this system.

The uumonitor command is helpful for detecting systems that have backlogs, have gone off the network for a while, have changed phone numbers, and so forth. The CNT: field is useful for detecting a system whose login or password has changed. If CNT: gets larger than the maximum allowable failures, 20 by default, no further attempts to connect will be made to this system.

If the number of C.files queued becomes unusually large, (depending on the system, from 100 to 1000) try to determine the cause of the backlog. If a separate subdirectory does not exist for the backlogged system, create the subdirectory and move the spooled files from the DEFAULT directory to the individual system subdirectory. When the problem is resolved, make an entry in /var/uucp/LIST.HOUR to help clear out the backlog. For more information on uumonitor, see uumonitor(8c) in the *ULTRIX Reference Pages*.

3.3.2 Obtaining the Status of the uucp Utility

The uustat command displays the status of previously specified uucp commands, cancels previously specified uucp commands, or provides general status on uucp connections to other systems.

The uux command requests are not recorded in the uustat logging files. Therefore, uustat does not record mail and news requests.

Some of the options are:

- mmch Report the status of accessibility of machine *mch*. If *mch* is specified as *all*, then the status of all machines known to the local uucp system are provided.
- kjobn Kill the uucp request whose job number is *jobn*. The killed uucp request must belong to the person issuing the uustat command, unless that person has superuser privileges.
- chour Remove the status entries which are older than *hour* hours. This option can only be executed by the user uucp or the superuser.
- uuser Report the status of all uucp requests issued by *user*.
- ssys Report the status of all uucp requests that are destined for remote system *sys*.
- ohour Report the status of all uucp requests that are older than *hour* hours.
- yhour Report the status of all uucp requests which are younger than *hour* hours.
- jall Report the status of all uucp requests or the specified job request number.
- v Report uucp status in words rather than code. If this option is not specified, a status code is printed with each uucp request.

When no options are given, the uustat command prints the status of all uucp requests issued by the current user. To focus on particular jobs, you can use options. For example:

```
# uustat -usteve -slimbo -y63 -v
```

This command prints the status, in words, of all uucp jobs issued by user *steve* that were destined for system *limbo* within the last 63 hours. The format of each job status entry is:

job# user destination spool_time status_time status

The status can be either an octal number or a verbal description.

The octal codes correspond to the following conditions:

Octal Code	Status
00001	The copy failed for unknown reasons
00002	Permission to access local file is denied
00004	Permission to access remote file is denied
00010	Bad uucp command is generated
00020	Remote system cannot create temporary file
00040	Cannot copy to remote directory
00100	Cannot copy to local directory
00200	Local system cannot create temporary file
00400	Cannot execute uucp
01000	Copy succeeded
02000	Copy finished, job deleted
04000	Job is queued

You can request the status of machine accessibility with the `-m` option. The format for the machine accessibility status entries is:

system status-time last-success-time status

<i>system</i>	The system in question.
<i>status-time</i>	The time the last status entry was made.
<i>last-success-time</i>	The time of the last successful connection to this system. Note that this does not imply that the entire session was completed, but rather that the transfer daemons successfully completed the handshaking phase and were able to begin transferring files.
<i>status</i>	A self-explanatory description of the machine status. See <code>uustat(1c)</code> in the <i>ULTRIX Reference Pages</i> for more information on using <code>uustat</code> .

3.3.3 Obtaining a Log of File Transfers

The `SYSLOG` file records the number, size and source of each data transmission. Each entry has this format:

uid remote_sys (date) (int_time) sent/rec_data bytes dur, Pk: n Rxmt n

<i>uid</i>	The effective user ID of the running process.
<i>remote_sys</i>	The name of the remote system where the data is sent or received.
<i>date</i>	The date of the transaction, including the time of day.
<i>int_time</i>	The internal representation of the date.
<i>sent/rec_data</i>	Indicates whether the data was sent to or received from the remote site.
<i>bytes</i>	The number of bytes transferred.
<i>dur</i>	The duration of the transfer in seconds.
<i>Pk:</i>	The number of packets needed to send the data.
<i>Rxmt</i>	The number of packets that had to be retransmitted.

3.4 Adding and Deleting Incoming Connections

If a remote site requests a uucp connection to your system, you need to update the information in your uucp-related files. To do this, run the `uucpsetup` command with the `-i` option:

```
# uucpsetup -i
```

Then answer the questions pertaining to incoming connections.

To deny a remote site access to your system, delete its entry from `/var/uucp/USERFILE`, and remove its login name from the `/etc/passwd` file. For example, to deny a system named `sysY` from accessing your system, delete the `sysY` entry from the `USERFILE`. Assuming the following is your `USERFILE`, you would edit `/var/uucp/USERFILE` to delete the third line:

```
remote, X1      /usr/spool/uucppublic
local,  X9      /
max,sysY c  /usr
max,sysZ      /sys
blimpy,sysQ    /
nuucp, /usr/spool/uucppublic
```

The presence of the `INSECURE` file makes your system potentially insecure. If the file `/var/uucp/INSECURE` exists, then all connection requests from systems not listed in `USERFILE` are allowed, with the remote system having the default parameters allotted by the `remote` entry in `USERFILE`, that is, an execution access level of 1.

Note that the remote system still has to log in to your system successfully even if the `INSECURE` file exists. Thus, it must have a valid uucp login name and password.

If the `INSECURE` file does not exist, your system is more secure because only those remote systems that have a specific entry in `USERFILE` can access your system.

3.5 Adding and Deleting Outgoing Connections

To allow your system to initiate a uucp connection with a remote site, you need to update the information in your uucp-related files. To do this, run the `uucpsetup` command with the `-o` option:

```
# /etc/uucpsetup -o
```

Then answer the questions pertaining to outgoing connections.

To prevent your system from initiating uucp connections with remote sites, delete the remote site's entry from `/var/uucp/L.sys`. For example, to prevent your system from connecting with `sysY`, delete the `sysY` entry from the `L.sys` file. Assuming the following is your `L.sys` file, you would edit `/var/uucp/L.sys` to delete the second line:

```
sysX Any ACU 1200 777-7777 "" \r\d ogin:-EOT-ogin: Ufoobaz ssword dog
sysY Any ACU 1200 222-2222 "" \r\d ogin: xuucp ssword frumples
sysZ Any ACU 300 465-1234 "" \r ogin:-EOT-ogin: Usys1 ssword testing
```

3.6 Adding, Deleting, and Moving Modems

To enable the `tip` command, you must set up the uucp modems, even if you will not be running the uucp utility.

To add a modem to your uucp facility, run the `uucpsetup` command with the `-m` option:

```
# uucpsetup -m
```

Then answer the questions pertaining to modems.

If you need to remove a modem or move one from one port to another, follow these steps:

1. Edit the `L-device` file. You may also need to edit the `L.sys` file, depending upon whether you have specific devices listed there. Delete or rename the corresponding device entries (dv) in these files as necessary.
2. Edit the `/etc/ttys` file and delete or modify the corresponding special file entries. For information about this file, see `ttys(5)` in the *ULTRIX Reference Pages*.
3. Activate the changes to the `/etc/ttys` file with the following command:

```
# kill -HUP 1
```
4. Edit the `/etc/remote` file and delete or rename the corresponding device entries (dv) for the `tip` utility. For further information, see `remote(5)` in the *ULTRIX Reference Pages*.
5. Move the appropriate `/dev` entries. For details about the `/dev` directory, see the *Guide to System Environment Setup*.
6. Physically move the modem cable from one port to another if you are moving the modem.

3.7 Adding Direct-Connect Lines

To add a direct-connect line, you need to modify several files on both the incoming and outgoing systems.

To set up direct-connect lines for the `uucp` utility on the outgoing system:

1. Edit the `/var/uucp/L.sys` file. Add an entry for the system to which you are connecting. For example:

```
tigger Any tty03 9600 tty03 "" \r ogin: poo ssword: bear
```
2. Edit the `/var/uucp/L-devices` file. Add an entry for the device that is the directly connected line. For example:

```
DIR tty03 tty03 9600 direct
```
3. Change the mode of the device. For example, if the line is `tty03`, type this command:

```
# chmod 666 /dev/tty03
```

To set up direct-connect lines for use with the `tip` command on the outgoing system:

1. Edit the `/etc/remote` file and add an entry for the device. For example:

```
tigger:dv=/dev/tty03:br=#9600:pa=none
```


2. Change the mode of the device. For example, if the line is tty03, type this command:

```
# chmod 666 /dev/tty03
```

For both the uucp and the tip utilities, edit the /etc/ttys file on the incoming system. Be sure that the entry for the incoming direct-connect line specifies the getty daemon. For example:

```
tty04 "/etc/getty std.9600" vt100 on nomodem
```

Make sure both incoming and outgoing systems are using the same baud rate.

3.8 Cleaning Up Excess Files and Directories

The uucp facility cleans up its temporary files and directories after a uucp connection has been completed. However, there are some files and directories that you must delete periodically, depending on the number of uucp connections your system makes.

One directory that tends to grow quickly is the /usr/spool/uucppublic directory. This directory grows because remote sites usually have access to it and can put files there. To control the size of this directory, you may need to delete files from it.

Another directory that tends to grow quickly is the /usr/spool/uucp directory. This directory contains all the log files recording your system's uucp activity. If any of the log files such as ERRLOG, SYSLOG, or LOGFILE becomes too large, you can delete them.

3.9 Maintaining System Security

System security is maintained when remote systems with uucp connections cannot access files on your local system. There are two ways to maintain system security:

- Assign the appropriate file access modes
- Assign the appropriate entries to the /var/uucp/USERFILE

You should remind all users to keep their individual files and directories properly protected. They can use the chmod command to change their files' modes.

You should also be sure that all system files have the correct protection modes. In its most insecure state, the uucp utility can access files that are readable by uucp. This includes any file with mode xx4. The uucp utility can overwrite files that are writable by uucp, that is, if the file modes are xx2 or xx6.

The USERFILE file is the first measure of system security. It provides the following levels of security:

- File access permission for remote and local users
- Login security
- Remote execution permissions

Each line in /var/uucp/USERFILE has the following format:

```
login,sys Xn c path-name path-name ...
```


<i>login</i>	The login name for a remote system or local user.
<i>sys</i>	The name of the remote system; optional.
<i>Xn</i>	The level of execution assigned to <i>sys</i> ; optional, except in the default entries <i>remote</i> and <i>local</i> .
<i>c</i>	The optional callback required flag.
<i>path-name</i>	A pathname prefix that <i>sys</i> can access.

3.9.1 File Access Security with USERFILE

The domain of accessibility of a remote system is restricted by */var/uucp/USERFILE*. An entry should exist for each system that defines which paths the system can access. If no entries exist for a particular system, the default entries are used.

The */var/uucp/USERFILE* must be set up with default entries that use this format:

```
remote, X# /some_path_name_for_remote_systems
local, X# /some_path_name_for_local_users
```

remote The default entry for remote systems that do not have an explicit entry in */var/uucp/USERFILE*. Do not be too liberal with this entry. A typical path allowed for remote users is:

```
remote, X1 /usr/spool/uucppublic
```

The */usr/spool/uucppublic* directory is a well known public repository. Users should not leave important files in this area.

local The default entry for local users. Usually, the normal ULTRIX system file access modes are all the security that is imposed on local users. Therefore, the typical default entry for local users is:

```
local, X9 /
```

X A keyword used by *uucp* to identify the set of commands that a remote system can execute on the local system. In the following examples, the *X* field is included for accuracy only. For information about security from remote systems, see Section 3.9.2.

Note

The default entries must be supplied; otherwise, the *uucp* utility fails.

In addition to the default entries, per-system entries may also be supplied. These entries provide the flexibility to give trustworthy systems less restrictive access to the local system. The following examples illustrate the alternatives:

Example 1:

```
remote, X1        /usr/spool/uucppublic
local, X9/
max,sysX        /usr/sources
```

This example allows remote system *sysX*, which has logged in to the local system with the login name *max*, to access anything that has the prefix */usr/sources*.

All other systems can access only the public directories.

Example 2:

```
remote, X1      /usr/spool/uucppublic
local,  X9/
max,    /usr/usr/src/share
```

This example allows any system or user who has logged in to the local system with login name max to access anything in or below the directories /usr and /usr/src/share. Note that several systems could log in with this same login name, so care should be taken to restrict access rights appropriately. All other systems can access only the public directories.

3.9.2 Login Security with USERFILE

The uucp utility tries to ensure that only legitimate systems log in to the local system. When a remote system logs in, it passes its name to the local system. The uucp utility crosschecks the system name and login name against the /var/uucp/USERFILE. If an entry exists for that system name and the login name does not match the one specified in the USERFILE, the connection attempt is terminated. The following example illustrates this situation:

Sample USERFILE:

```
remote, X1      /usr/spool/uucppublic
local,  X9/
max,sysY c/usr
max,sysZ        /sys
blimp,sysQ/
nuucp,          /usr/spool/uucppublic
```

Case 1:

The competition across the street has unscrupulously obtained the login name blimp and the password for your uucp utility. They successfully connect to the local uucp. The local uucp utility then checks for a USERFILE entry for blimp. It finds the entry and knows that only sysQ can connect with the login name blimp. Since the name of the remote system is syszero, the connection attempt fails.

Case 2:

In this case, the same competition stole the login entry for nuucp. Since no system name is provided in the USERFILE for the nuucp entry, the connection attempt succeeds.

Case 3:

It may happen that a system logs in and there is no entry in USERFILE that corresponds to either the login name or the remote system name. The uucp utility handles this situation as follows:

If the file /var/uucp/INSECURE exists, the connection request is accepted.
If /var/uucp/INSECURE does not exist, the connection request is rejected.

This option is provided so that you do not need to supply an entry for every system that can log in. The default (remote) entry is used for systems that do not have an entry in the USERFILE, for example, when the system manager only wants to worry about two uucp passwords: one login for trustworthy systems and another login for the other systems, for example USENET.

Here is a sample USERFILE:

```
remote, X1      /usr/spool/uucppublic
local,  X9/
safeuucp, /
unsafeuucp,     /usr/spool/uucppublic
```

Since no system names are specified, connection attempts that use the login names safeuucp or unsafeuucp succeed. Attempts to connect with other login names succeed only if /var/uucp/INSECURE exists, and then they only receive the security level of the default path.

Case 4:

If you believe it is possible to forge remote system names, you should use the callback option. In the USERFILE example, if a successful login is made from a system that claims to be sysY, the uucp utility rejects the request and then tries to call the real sysY. This is the most secure form of USERFILE entry. Note that if both the destination and local system use the callback option, an infinite loop of call backs can occur. Make sure this does not happen.

3.9.3 Remote Execution Security

You can restrict remote execution to specified systems only. In addition, systems that are allowed to execute commands on the local system can be limited to a subset of allowable commands.

The X# field of a USERFILE entry is used to provide levels of security. The # can range from 0 to 9, where 0 is the most secure level and 9 is the least secure level. When the execution daemon (uuxqt) processes a remote execution request, it obtains the security level from the USERFILE that corresponds to the remote system making the request. The command to be executed also has a level number. If the level associated with the remote system is greater than or equal to the number associated with the command, then the uuxqt daemon grants permission to execute that command. Otherwise, the remote execution request fails.

Note

You must specify the execute field in the default USERFILE entries. Otherwise, the uucp utility fails. No other USERFILE entries need to have the execute level specified. For entries that do not have an execution level listed, the default execution level is used for that system. The default execution level for remote systems is obtained from the remote USERFILE entry. The local USERFILE entry provides the execution level to local users.

Execution levels are provided on a system-name basis only, not to particular users. You cannot use a USERFILE entry that specifies a login name, but no system name, to determine the execution level of a system that logs in with this entry. Instead, machines that do not have their own USERFILE entry will use the default remote USERFILE entry.

The next example illustrates remote execution security, assuming the following USERFILE and L.cmds file.

Here is the sample /var/uucp/USERFILE:

```
remote, X0      /usr/spool/uucppublic
local,  X9/
maxuucp,sysmax X3  /usr
```



```
xuucp,systemx X1 /usr/spool/uucppublic
yuucp,systemy X1 /usr/spool/uucppublic
zuucp,systemz /usr/spool/uucppublic
nuucp, /usr/spool/uucppublic/stockroom
ruucp, X5 /usr/spool/uucppublic/stockroom
```

Here is the sample `/var/uucp/L.cmds` file:

```
rmail X1
rnews X1
uusend X2
```

Explanation: The default remote entry prevents any system that does not have its own `USERFILE` entry from executing any of the commands in `L.cmds`. They can, however, send or receive files from the public directories, and the systems `sysmax`, `systemx`, and `systemy` can execute `rmail` and `rnews`. Yet `sysmax` is the only remote system that can execute `uusend`. Any system that logs in as `nuucp` is provided the default execution level, which in this case is 0, and cannot execute any commands.

The important point to notice is that these latter systems that log in as `nuucp` are not provided with the security level of the default path. Therefore, the systems that log in as `nuucp` can only send and receive files to and from `/usr/spool/uucppublic/stockroom`.

Any system that logs in as `ruucp` has the same execute permissions as those that log in as `nuucp`. Because no system name is provided in the `ruucp` `USERFILE` entry, the system ignores the `X` field and uses the default execution level instead.

Local users can access all of the commands in `L.cmds`.

3.10 uucp Commands

See the following `uucp` commands in the *ULTRIX Reference Pages*: `uucp(1)`, `uulog(1)`, `uuname(1)`, `uustat(1)`, and `uux(1)`.

This chapter provides information on how to configure the MICOM Micro800/X.25 Packet Assembler Disassembler (PAD) for the `tip` and `uucp` ULTRIX software utilities, and discusses the following topics:

- Connecting the PAD
- Setting up the `uucp` utility for the PAD
- Setting up `tip` for use with the PAD

The MICOM Micro800/X.25 PAD hardware is not sold or supported by Digital Equipment Corporation.

4.1 Description of the MICOM PAD

The MICOM Micro800/X.25 PAD is sold and supported by MICOM Systems Inc. The PAD implements the Consulting Committee for International Telephony and Telegraphy (CCITT) recommendations X.3, X.28, and X.29. It connects directly to an X.25 access line (also called a trunk) and performs the necessary operations to place calls and create connections to other data terminal equipment (DTE) on the X.25 network. Asynchronous devices such as terminals or multiplexer lines gain access to the PAD by being connected to a cable to one of its channels.

Depending on the model purchased, the PAD has from 4 to 16 channels for sending or receiving X.25 calls. By attaching the PAD channels to terminal multiplexer lines on a processor running ULTRIX software, you can configure the PAD to support outgoing `uucp` connections, outgoing `tip` connections, and incoming `uucp` and login services.

Although each channel can be configured for more than one purpose, in this discussion assume that each channel is configured for a specific purpose. For example, channel 1 is for outgoing `uucp` calls, channel 2 is for outgoing `tip` calls, channel 3 is for incoming `uucp` connections, and channel 4 is for incoming login service.

4.2 Connecting the PAD

Before connecting multiplexer lines to PAD channels, follow these steps:

1. Decide which channels to dedicate for which purposes, and then reserve that many multiplexer lines on your ULTRIX operating system. These lines must support modem control.
2. Turn off any `getty` processes that may be running on the dedicated multiplexer lines:
 - a. Edit the `/etc/ttys` file to make sure that the status of the dedicated lines is off.

- b. Send a HUP signal to the `init` process by typing:

```
# kill -HUP 1
```

The HUP signal makes the changes in the `/etc/ttys` file take effect.

The multiplexer lines can now be connected to the PAD channels using Digital RS-232 modem-compatible cables.

Before setting up the PAD, read the appropriate manual written for the MICOM Micro800/X.25 Concentrator PAD to become familiar with the device. The PAD documentation contains the necessary information for configuring and using the PAD. If you are setting up the PAD for the `uucp` or `tip` utilities, first acquaint yourself with the PAD by using it with a terminal until you are able to place calls successfully and create connections to remote devices. Then review the sections in the PAD documentation for information on the specific configuration parameters to make the PAD work with the `uucp`, `tip`, and remote login utilities.

The PAD makes provisions for placing calls between its channels as a form of loopback. With a loopback, calls can be made without ever going over the X.25 network.

For example, suppose you have connected terminals to channels 3 and 4. To place a call from channel 3 to channel 4, type the following at channel 3:

```
C 00/04
```

The DTE address 00 places calls between channels.

4.3 Setting Up the `uucp` Utility for the PAD

This section describes how to set up the `uucp` utility for use with the PAD. This section discusses:

- Setting up incoming `uucp` lines
- Setting up the `uucp` utility to dial out over the PAD

The `uucp` utility has an additional protocol called `f-proto`, which is much faster than the standard `g-proto` protocol. The calling machine initiates the `f-proto` protocol; however, in order for it to be effective, both machines must recognize the protocol. Furthermore, the PAD parameters shown in the profile below must be set for both the PAD channel placing the call and the PAD channel receiving the call.

A device profile is a list of PAD parameter values. When a device profile is assigned to a channel, the channel's PAD parameters take on the values listed in the device profile. You can define up to eight device profiles, numbered from 1 to 8. Reserve a device profile number for `uucp` and set the profile as shown below. For the following example, assume profile 8 was chosen for `uucp`.

You should assign the same device profile to all channels dedicated for `uucp`, for example, profile 8. Profile 8 needs the following PAD parameters set:

<code>x28acc=0</code>	<code>bits/char=3</code>
<code>echo=0</code>	<code>dv_parity=0</code>
<code>data_fwd=2</code>	<code>net_parity=0</code>
<code>idletimer=10</code>	<code>c.xon=17</code>
<code>dv_flow=1</code>	<code>c.xoff=19</code>
<code>s.signal=0</code>	<code>special_flow=0</code>
<code>break=0</code>	<code>count_fwd=0</code>
<code>cr_pad=0</code>	<code>escdelay=0</code>


```

l.fold=0          c.break=0
speed=14          c.suppress=0
p.flow=1          c.subs=0
autolf=0          echoctrl=0
lf_pad=0          special_echo_id=0
edit=0            pagelength=0
c.del=0           c.page=0
l.del=0           ff_pad=0
l.disp=0          inactivity=0
dv.type=2         x29acc=0
echomask=0

```

Note that the speed parameter selected is 9600 baud (value 14). Refer to the MICOM documentation for information on how to change the speed parameter.

4.3.1 Setting Up Incoming uucp Lines

To set up incoming uucp lines, configure the channel dedicated for incoming uucp calls as follows:

```

Profile: 8
Call option: 1
Address id Mnemonic: **
Channel Option: 6

```

Then follow these steps:

1. Set the profile selection to the number assigned previously. The number 8 is used in this example because the profile was assigned 8 in the previous section. Be sure that the channel option is 6. Channel option 6 raises the carrier detect signal each time a call comes in. Be sure the carrier detect signal is passed through the cable to the multiplexer line. The situation is analogous to having a dialup modem installed. Leave the other factory-set values alone.
2. Place a getty process on the line. To do this, edit the /etc/ttys file and change the status of the line to on modem. Then, send a hangup signal to init by typing:

```
# kill -HUP 1
```

Assume line ttyh6 is connected to the PAD channel 3 and is reserved for uucp incoming connections. The following is an example entry for the /etc/ttys file:

```
ttyh6 "/etc/getty T9600" vt100 on modem # PAD chan 3
```

The result is a PAD channel that can be called over the X.25 network by another system running the uucp utility to request the f-proto protocol.

3. Configure the channel reserved for outgoing uucp connections as follows:

```

Profile: 8
Call option: 1
Address id Mnemonic: **
Channel Option: 0

```

The channel option 0 means that the channel is dedicated.

4.3.2 Setting Up uucp to Dial Out Over the PAD

After you have configured the PAD channels for uucp dialout, you are ready to set up the uucp utility. For information on how to set up the uucp utility, first see Chapter 2. Then follow these steps:

1. Edit the `/var/uucp/L-devices` file to reflect the addition of the PAD dialout lines. Here is the new format for the L-devices file:

type line call-unit speed brand preferred_proto

The PAD channels are treated as direct lines. The L-devices file now contains the *preferred protocol* field from which you request f-*proto* for lines connected to PAD dialout channels. For example, suppose the PAD channel for uucp dialout is connected to `ttyh5`. The proper L-devices entry is:

```
DIR ttyh5 ttyh5 9600 direct f
```

Suppose you have two channels configured for uucp dialout: one connected to `ttyh7`, and the other to `ttyh8`. Here are the proper entries:

```
DIR ttyh7 ttyh7 9600 direct f
DIR ttyh8 ttyh8 9600 direct f
```

2. Edit the `L.sys` file. This file holds entries to call other systems. It contains the information necessary to call the host and perform the login sequence. To call a host over the X.25 network, assume that it is on a direct-connect line attached to the dialout channel of the modem. Include the PAD commands to call the host as part of the login sequence. For example, suppose the name of the system you are calling is `mozart`, the X.25 DTE address is `12345`, and the PAD channel you are calling `mozart` on is channel 3, the one dedicated for uucp login. Your PAD uucp dialout channel is connected to `ttyh5`, the login name is `Umozart`, and the password is `donny`. The following `L.sys` entry could be used to call system `mozart`:

```
mozart Any DIR 9600 ttyh5 "" \r "" C\s12345/03 ogin:-\r-ogin: \
Umozart ssword: donny
```

In this example, the entire entry is on one line, although it may wrap around. The information that places the call over the PAD is `C\s12345/03` and is interpreted as follows:

- C The PAD command to place an X.25 call.
- \s Indicates how to include a space character in an `L.sys` entry. Make sure you have a space between the PAD command (C) and the DTE address (12345).
- 12345 The DTE address.
- /03 Address channel 3 on the called PAD.

The rest of the line is the login sequence. Note that the device containing the PAD dialout line is wired into the `L.sys` entry for calling system `mozart`. Even if two PAD channels are reserved for uucp dialout, the same one is always used to call `mozart`. Rotary action is not possible.

4.4 Setting Up tip for Use with the PAD

This section describes how to set up the `tip` utility for use with the MICOM PAD, and discusses the following topics:

- Setting up outgoing `tip` connections
- Setting up incoming login service

4.4.1 Setting Up Outgoing tip Connections

The PAD `tip` dialout lines are treated like direct connects. For example, if you are on the X.25 network called `telenet` and your outgoing `tip` channel is connected to `/dev/ttyh7`, the proper entry for the `/etc/remote` file is:

```
telenet|pad|PAD outgoing channel:\
      dv=/dev/ttyh7:br#9600
```

By typing the following, you make a connection to the PAD:

```
# tip telenet
```

You can then place your X.25 call just as if you were connected directly to the PAD, which in effect you are.

If you have three outgoing `tip` channels connected to `/dev/ttyh4`, `/dev/ttyh5`, and `/dev/ttyh6`, then you should have the following entry in the `/etc/remote` file:

```
telenet|pad|PAD outgoing channel:\
      dv=/dev/ttyh4,/dev/ttyh5,/dev/ttyh6:br#9600
```

This causes a rotary effect. If the first channel is busy, `tip` tries connecting to the next channel, and so on.

When you have completed your call, remember to type tilde-dot (~.) to make `tip` hang up the connection. The `tip` command has no other way of knowing when your call is over.

The PAD parameters specify information such as whether the PAD echoes characters and when the host PAD forwards characters it receives to the remote PAD. The standard convention for `tip` is that the remote host echoes characters. This convention is necessary for screen-oriented programs such as `vi`. However, it produces a high X.25 traffic rate of up to two X.25 packets for each character typed and, accordingly, causes high public data network (PDN) charges. The other extreme is to have the PAD perform all the echoing and editing features, and then only forward characters when, for example, an entire line has been typed. However, many programs do not function properly if they have to wait for an entire line before they can receive any characters.

There is no solution that fits all situations. Yet, if you use the following device profile, which causes high network charges, you do not lose any ULTRIX software functionality:

<code>x28acc=0</code>	<code>echomask=0</code>
<code>echo=0</code>	<code>bits/char=3</code>
<code>data_fwd=127</code>	<code>dv_parity=0</code>
<code>idletimer=10</code>	<code>net_parity=0</code>
<code>dv_flow=1</code>	<code>c.xon=17</code>
<code>s.signal=5</code>	<code>c.xoff=19</code>
<code>break=0</code>	<code>special_flow=0</code>
<code>cr_pad=0</code>	<code>count_fwd=0</code>
<code>l.fold=0</code>	<code>escdelay=0</code>


```

speed=14      c.break=0
p.flow=1      c.suppress=0
autolrf=0     c.subs=0
lf_pad=0      echoctrl=0
edit=0        special_echo_id=0
c.del=0       pagelength=0
l.del=0       c.page=0
l.disp=0      ff_pad=0
dv.type=2     inactivity=0
              x29acc=0

```

For purposes of this example, assume that profile 5 has been assigned the values shown above. Profile 5 forwards X.25 packets for every character typed. Although the PAD displays the PAD service prompt (`s.signal=5`), it does not echo the characters typed as you place the call because of the `echo=0` parameter.

Note

Remember, this profile (profile 5) may not be proper for your installation. The PAD parameter information in your PAD documentation gives other examples where the PAD is used to echo the characters and perform editing.

For outgoing tip connections, configure the reserved channels as follows:

```

Profile: 5
Call option: 1
Address id Mnemonic: **
Channel Option: 0

```

Note that the parameter for Profile is the only variable. The other parameters must have the values shown.

4.4.2 Setting Up Incoming login Service

The same issues related to outgoing tip connections are applicable with incoming login service. Follow these steps:

1. Set up the profile. Use the same profile for incoming login services that you use for outgoing tip connections. Configure the channels dedicated for incoming login calls as follows:

```

Profile: 5
Call option: 1
Address id Mnemonic: **
Channel Option: 6

```

The important setting is the channel option. Channel option 6 specifies that the channel will raise carrier detect when a call comes in. Thus, be sure that the carrier detect signal is passed through the cable to the multiplexer line. The situation is analogous to having a dialup modem installed.

2. Place a `getty` process on the line. First, edit the `/etc/ttys` file and change the status of the line to be on modem. Then, send a HUP signal to `init` by typing:

```
# kill -HUP 1
```

Assuming line `ttyh6` is connected to the PAD channel 3 and is reserved for incoming login connections, the following is an example entry for the `/etc/ttys` file:

```
ttyh6  "/etc/getty T9600" vt100  on  modem  # PAD chan 3
```


When a call comes over the X.25 network for a channel designated for incoming login service, the channel raises the carrier detect signal, the getty waiting on the line wakes up, and a login message appears for the user placing the call.

THE UNIVERSITY OF CHICAGO
LIBRARY
540 EAST 57TH STREET
CHICAGO, ILL. 60637

This chapter discusses how to troubleshoot the uucp utility.

If your system uses the supported hardware, most problems should be either hardware or administrative failures, such as files that are not set up correctly or wrong phone numbers. The discussion in this chapter should help determine the source of the problem.

The following administrative files contain uucp statistics to use for diagnosing problems:

- /usr/spool/uucp/LOGFILE
- /usr/spool/uucp/ERRLOG
- /usr/spool/uucp/SYSLOG

When a connection to a remote system does not seem to be working, look at the LOGFILE and ERRLOG files. They will help you determine if the problem is in the dialing stage, the login/handshaking stage, or the file transfer stage, and whether it is a hardware problem.

The most common errors occur when the remote site has set up its USERFILE incorrectly. An incorrect USERFILE results in messages being sent back to local users saying that remote access to path files is denied. However, at any time the remote system could have changed its password or phone number. You should contact the system manager of the remote system for updated information.

When error messages constantly refer to one ACU out of many, the problem is probably a bad ACU. If the SYSLOG file has recorded a lot of retransmissions, the hardware may be faulty, the communications line may be noisy, or the baud rate may be too high for the transmission media.

If the source files are available, you should try initiating a conversation with the remote system in question. To initiate a conversation, start a uucico process in the master mode in this format:

```
# /var/uucp/uucico -r1 -xn -Xn -ssystem
```

- | | |
|---------|---|
| r1 | Puts the uucico process into the master role. |
| xn | The debugging level. The number can be from 0 to 9. The higher the number, the more debugging output. No packet-level debugging is printed. |
| Xn | Obtains packet-level debugging output. The number can be from 0 to 9. The higher the number, the more packet-level debugging output. |
| ssystem | The system to be contacted. |

Another option to the uucico process is -f. This forces the uucico process to start a conversation with a specified system, regardless of any previous connection status as provided by the STST. files.

Even if the source code is not available, you can obtain useful information. The output of the `uucico` process follows the progress of the conversation. Debugging output from the slave `uucico` process is placed in the file `AUDIT` in the spool directory at the remote site. The output tends to be less meaningful than the `LOGFILE`, unless the source code is available to help interpret the messages. A detailed explanation of the messages is not given here because the messages are likely to change from version to version.

5.1 Transmission Messages in the LOGFILE

The `LOGFILE` contains information logged by the transfer process `uucico` concerning a conversation with a remote site. If a problem occurs during the connection stage, it is noted in this file. The entry format is:

login_name (time of transaction-process_number) message

5.1.1 Problems During the Connection Stage

You might see the following error messages if a problem occurs during the connection stage:

LOGIN FAILED

The `uucico` process got a carrier but could not log in.

FAILED (call to some system)

The connection attempt failed as a result of a previous message.

NO CALL (RETRY TIME NOT REACHED)

You attempted to call a system too soon after previous failed attempt.

CAN NOT CALL (SYSTEM STATUS)

The call to a remote system aborted because of a previous connection status, such as too many failed attempts to log in to the remote system or attempting to log in too soon after a previous failed connection attempt. Connection status information is kept in `/usr/spool/uucp/STST`. There is a separate `STST` file for each remote system. If the `STST` file corresponding to the remote system is removed, then a call is permitted to that remote system immediately.

NO CALL (MAX RECALLS)

The local system has failed to log in to the remote system the maximum number of times, which is 20 by default. No further attempts can be made by `uucico` until the `STST.remote` file is removed from the `/usr/spool/uucp/STST` directory.

SHARED LINE IN USE

You attempted to use a shared line that is busy.

ALREADY OPEN (device name)

This also gets the log message, direct line is already in use, except this message is logged for ACUs.

FAILED (HSM no carrier)

The carrier was not detected when using a Hayes Smartmodem.

df02/df03/df112 illegal return (FAILED acu=/dev/cua2, char=0)

This message indicates that the specified ACU was in a strange state, resulting in an unknown return character. This problem usually goes away when the current `uucico` process exits.

WRONG TIME TO CALL (*systemname*)

An attempt was made to call a system at a time outside the range specified in the `L.sys` file.

NO DEVICE

An attempt to call a system was made, but no ACU or hard-wired line devices were available.

using device (*device*, *fd*=#)

The `uucico` process is using *device* to call a remote system. The *fd* is the file descriptor that corresponds to *device*.

LOCKED (call to *systemname*)

An attempt to call a system was made for which a conversation was already in progress. Therefore a lock file, for example, exists for that system in the spool directory.

FAILED (LOGIN VS MACHINE)

A remote system tried to log in, but it failed the `USERFILE` security test.

TIMEOUT (DIALUP DN write)

There was no carrier detect after dialing a number.

FAILED (DIALUP ACU write)

An error occurred while writing the phone number to the ACU. If this error consistently occurs, it could be the result of an ACU hardware failure. There might also be something wrong with the mode of the special device file.

TIMEOUT (*systemname*)

A remote system has initiated a connection attempt and then stopped communication (reason unknown to the local site).

The following messages can occur at any time:

ret (#) from *system*!*user* (MAIL FAIL)

The `mail` command failed. The exit status is indicated by `ret`. The originator is *user* on the remote node: *system*.

cmd: *command*; ret: *signal* #, exit # (CMD FAILED)

A remote execution request failed. The *command* is the command that failed. The *signal* is the signal that caused the command to fail. The *exit* is the value returned by an `exit` system call in *command*.

XQT DENIED (*command*)

A remote system tried to execute *command*, but the request was denied by the local system.

cmd xqt'ing > 55 minutes (touch lock file)

The `uuxqt` daemon has been executing a command for 55 minutes. The lock files associated with this command are refreshed so that they will not be removed by another `uuxqt` process.

command terminated - exceeded time limit (*command*)

A command that is being processed by the `uuxqt` daemon has been running longer than approximately eight hours. The *command* is assumed to be a runaway process and is terminated.

system_name (execute level too low)

The *system_name* was not allowed to execute a command because it did not have a high enough execute level. The specific command should be named by a subsequent LOGFILE entry.

CAUGHT (SIGNAL #)

A signal was caught by *uucico* that caused it to terminate.

intrEXIT (signal: #)

A signal was caught by *uucico*. This signal was probably the result of damaged software, an undetected bug, or some other system problem.

hayes: closing (fd = #)

The *uucico* daemon has finished using a Hayes Smartmodem. The *fd* is the file descriptor associated with the ACU.

closed generic acu

The *uucico* daemon finished using the generic dialer.

These messages occur during the file transfer stage after the connection has been made:

FAILED (CAN'T READ DATA)

A work file (*C.file*) has specified a file to transfer, but that file is not readable by *uucp* or does not exist.

FAILED (CAN'T READ SPOOLED DATA)

A work file (*C.file*) has a reference to a spooled data file (*D.file*), and the spooled file is either unreadable by *uucp* or does not exist.

NOINPUT (set no incoming files allowed (*remote name*))

The file */var/uucp/NOINPUT* exists. The existence of this file is used as a flag to the *uucico* daemon: if *NOINPUT* exists, do not allow incoming files.

ACCESS (DENIED)

An attempt was made by a remote system to access a file for which it does not have USERFILE permission.

REQUEST

The remote system might have run out of space. An attempt was made to write to a directory that was not writable by *uucp*, or some USERFILE restriction was violated.

DENIED (CAN'T OPEN)

A remote site tried to receive a file that the local *uucico* daemon cannot access.

BAD READ# (expected *char* got *something_else*)

The local *uucico* daemon was waiting for a reply from the remote system, but either the return message was corrupted or the remote process did not reply.

FAILED (CAN'T CREATE TM)

An attempt to create a temporary file failed, which may indicate that the system is running out of space.

FAILED (conversation complete)

A conversation with a remote system stopped before all of the spooled files were transferred. The reason for the failure is not known, but it could have

been caused by something or someone killing the remote process or possibly by a lost line. If a conversation has lasted about 90 minutes and the remote site is running an older version of uucp, the problem may have been the premature removal of a LCK.file at the remote site.

The following informative messages can appear in the LOGFILE:

REQUEST (*char srcname dstname owner*)

A request to transfer a file to a remote site was made. If no followup message was made to indicate some kind of failure, the request was successful.

The *char* is the type of request: S for send, R for receive, X for remote execution.

The *srcname* is the name of the file on the local system.

The *dstname* is the name the file will have on the remote system.

The *owner* is the owner of the file.

REQUESTED (*char srcname dstname owner*)

A remote site has requested to transfer a file to the local site. If there is no subsequent failure message, then the transfer was successful.

OK (startup)

The local and remote sites have successfully agreed upon what low level packet protocol to use to transfer raw data.

OK (conversation complete)

All transactions with a remote system completed successfully.

uucp XQT (PATH.....;cmd)

A request from a remote site to execute *cmd* was successful.

XQT QUE'D (*cmd*)

A command (*cmd*) to be executed at a remote site was spooled.

:QUE'D

A user request to transfer a file was spooled.

5.2 Obtaining a Log of Administrative Errors

The ERRLOG file contains error messages that are less likely to occur during the normal operation of uucp. If a uucp support file such as the USERFILE is improperly set up, the problem is noted here in the ERRLOG file. If administrative problems such as this occur, the software will most likely stop executing. Therefore, it is important that administrative problems listed in the ERRLOG be corrected quickly. If the system has run out of space preventing the creation of files, or if referenced files do not exist, the problem is noted here. Problems that occur during the transmission of raw data packets are also noted here. Packet problems occur infrequently and do not halt the uucico daemon. They are more indicative of a poor connection or an over-loaded system and sometimes may point to a problem in the hardware.

The format of an ERRLOG message is:

ASSERT ERROR (*process name*) *process# time_of_entry message return_code*

ASSERT_ERROR

Redundant information indicating that an ASSERT ERROR has occurred.

process_name

The name of the process in which the error occurred. It could be `uucico`, `uucp`, `uuxqt`, `uux`, `uustat` or any other `uucp`-related program.

process#

The process identification number.

time_of_entry

The time the ASSERT error occurred.

message

Indicates the nature of the error.

assert_return_code

A returned value that can be helpful for finding the source of the error.

Some typical ASSERT messages are:

PKassert

Any message that begins with PK comes from the packet transmission software. It is most likely due to a noisy line or a failure at the remote system.

NO default entry for remote machines

The `USERFILE` does not have the default entry for remote systems.

NO default entry for local users

The `USERFILE` does not have the default entry for local users.

xeq level undefined in `USERFILE`

The remote execution security level (X#) was not specified for a default `USERFILE` entry.

CAN'T OPEN *filename*

The named file probably does not exist.

WRONG ROLE

During the file transfer stage, the `uucico` process reversed roles from master to slave or slave to master at the wrong time.

ARG COUNT<#

This message indicates that a work file (`C.file`) has been corrupted.

STAT FAILED *filename*

The `stat` system call failed. If it happens continuously, the named file is probably missing, but should not be.

BAD LINE

A bad entry in the `L-devices` file has been encountered.

TOO MANY SUBDIRECTORIES

The maximum number of individual system subdirectories has been created. No more can be made.

TOO FEW LOG FIELDS

The login/expect sequence of the `L.sys` entry for the remote system is incorrect.

BAD SPEED

The desired line speed is not allowed.

RETURN FROM STTY

An attempt to set the terminal I/O parameters of the outgoing line has failed.
This could be either a hardware or a system problem.

BAD WRITE genbrk#

An error occurred while generating a BREAK character on the outgoing line.

BAD DIRECTORY *directory_name*

The named directory does not exist or is not set to the correct modes.

CAN NOT GET sequence file lock

A lock file cannot be created so that the sequence file
/usr/spool/uucp/sys/*sysname*/SEQF can be accessed.

PERMISSION DENIED (Incoming C. file)

The uucico daemon does not permit work files (C. files) to be transmitted to
the local site because of security reasons.

1900

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

Index

A

AUDIT file, 5-2

C

callback option

 paying phone bill, 2-3n

configuration file (sendmail)

 refreezing, 2-3

connection stage

See also handshaking stage

 problems, 5-2

 problems with, 5-2

crontab file

 typical uucp entries, 3-2

D

device profile

 defined, 4-2

dial code

 file set-up, 2-8

dialing stage

 problems with, 5-1

directory

 required, 1-3

E

ERRLOG

 troubleshooting uucp, 5-1

ERRLOG file

 contents, 5-5

 message format, 5-5

 message list, 5-6 to 5-7

error messages

 ACU, 5-1

 ASSERT, 5-6 to 5-7

F

file

 required, 1-3

 transferring, 1-1

H

handshaking stage

 problems with, 5-1

hardware

 required, 1-2

I

INSECURE file

 system security, 3-6

L

LCK file

 creating, 1-2

L.cmds file

 entry format, 2-9

L-devices file

 editing, 44

 entry format, 2-8

 typical entries, 2-9e

L-dialcodes file

 entry format, 2-8

lines

 adding direct-connect, 3-7

log

file transfers, 3-5

LOGFILE

entry format, 5-2

message list, 5-2 to 5-5

troubleshooting uucp, 5-1

login stage

problems with, 5-1

L.sys file

editing, 2-8n, 4-4

setting up, 2-5 to 2-8

M

MICOM PAD

connecting, 4-1

description, 4-1

parameters, 4-2, 4-2

modem

adding for uucp, 3-6

changing for uucp, 3-7

deleting for uucp, 3-7

modem line

configuring manually, 2-4

modems

tip utility, 1-3n, 2-1n

uucp utility, 2-2n

O

option

channel, 4-3

defined, 4-3

profile, 4-3

P

PAD

See MICOM PAD

parameters

default for USERFILE, 3-6

profile, 4-2

password file

setting up, 2-5

password file (cont.)

user ID, 2-5n

profile

MICOM PAD

See parameters

R

refreezing sendmail file

single-user mode, 2-3

remote system file

See L.sys file

S

spool directory

required for uucp, 2-10 to 2-12

SYSLOG file

entry format, 3-5

system security

uucp, 3-8 to 3-12

T

tip command

enabling, 3-6

setting up for PAD, 4-5 to 4-7

transfer stage

problems with, 5-4

transmission log

obtaining, 5-2 to 5-5

trunk

See X.25 access line

ttys file

editing, 4-6

U

USERFILE

format, 3-8

login security, 3-10 to 3-12

restricting remote execution, 3-11

specifying execute field, 3-11n

uucico program

defined, 1-1

uucico program (cont.)

troubleshooting a connection, 5-1

uucp

defined, 1-1

overview, 1-1

uucp utility

adding dialout PAD lines, 44

adding individual subdirectories, 212

adding outgoing connections, 3-6

adding remote connections, 3-6

administrative files, 5-1 to 5-7

compacting spool subdirectories, 3-3

creating individual subdirectories, 2-10 to 212

defined, 1-1

deleting excess files, 3-8

deleting outgoing connections, 3-6

deleting remote connections, 3-6

diagnosing problems, 5-1 to 5-7

installing subdirectories for new systems, 2-10

installing subdirectories on old systems, 2-11

logging administrative errors, 5-5

maintaining, 3-1 to 3-12

monitoring, 3-3 to 3-5

polling remote sites, 3-1

setting up automatically, 2-1 to 2-3

setting up for PAD, 4-2 to 4-4

setting up incoming lines, 4-3

setting up manually, 2-4 to 212

setting up outgoing tip connections, 4-5

troubleshooting, 5-1 to 5-7

uucpsetup command

running, 2-1 to 2-3

uumonitor program

defined, 3-3 to 3-4

output format, 3-3

uustat program

defined, 3-4 to 3-5

job status codes, 3-4

job status entry format, 3-4

options, 3-4

output format, 3-4

uux, 3-4

uux program

defined, 1-1

uuxqt program

functions, 1-2

V**/var/uucp/uumonitor file**

See uumonitor program

X**X.25 access line**

connecting to, 4-1

Y**YP service**

uucp utility, 2-1n

uucpsetup command, 3-1n

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

1000

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-baud modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital Subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	_____	Local Digital subsidiary or approved distributor
Internal*	_____	SSB Order Processing - WMO/E15 or Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

* For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

ULTRIX
Guide to the uucp Utility
AA-MF03B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:

	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness (enough information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page layout (easy to find information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

What would you like to see more/less of? _____

What do you like best about this manual? _____

What do you like least about this manual? _____

Please list errors you have found in this manual:

Page	Description
------	-------------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

Additional comments or suggestions to improve this manual:

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

----- Do Not Tear - Fold Here and Tape -----

digital™



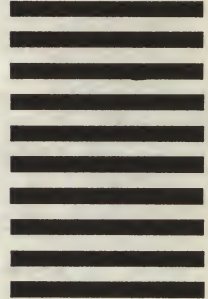
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3-2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062-9987



----- Do Not Tear - Fold Here -----

Cut
Along
Dotted
Line

Reader's Comments

ULTRIX
Guide to the uucp Utility
AA-MF03B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:

	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness (enough information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page layout (easy to find information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

What would you like to see more/less of? _____

What do you like best about this manual? _____

What do you like least about this manual? _____

Please list errors you have found in this manual:

Page	Description
------	-------------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

Additional comments or suggestions to improve this manual:

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

----- Do Not Tear - Fold Here and Tape -----

digital™



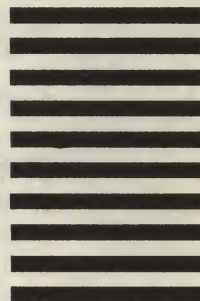
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3-2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062-9987



----- Do Not Tear - Fold Here -----

Cut
Along
Dotted
Line

digital